CLAIMS

We claim:

1. In combination, a communications interface between a local computer and a remote system having a graphical user interface; a scripting language; and graphical user interface language extensions that enable said scripting language to control said remote system responsive to images appearing on said remote system graphical user interface.

2. The combination of claim 1 further comprising a display of said remote system screen at said local computer.

3. The combination of claim 2 further comprising a command capture interface for generating graphical user interface language extensions commands from actions performed upon said local computer display of said remote system screen.

4. The combination of claim 3 further comprising a means for transferring said graphical user interface language extensions commands to a script responsive to said actions performed upon said local computer display of said remote system screen.

5. The combination of claim 1 further comprising a graphical user interface recognition subsystem which detects image components appearing upon said remote system graphical user interface and produces an image detection signal responsive thereto.

6. The combination of claim 5 further comprising an initiate action subsystem which evokes a remote system action responsive to said image detection signal.

7. The combination of claim 1 wherein said communications interface further comprises a means for communicating keyboard actions to said remote system, a means for communicating mouse actions to said remote system, and a means for communicating screen updates from said remote system to said local computer.

8. A method for remotely testing the operation of a computer system, comprising the steps of:

receiving a first element of said computer system graphical user interface;

generating a user input action within said computer system responsive to said first element;

monitoring said computer system graphical user interface for an expected second element within a predetermined time interval; and

signaling a failure if said predetermined time interval elapses without detecting said expected second element.

9. The method of claim 8 wherein further comprising the steps of:

transferring said user input action to a stored script;

re-executing said steps of receiving, generating, monitoring and signaling subsequent to said storing step under control of said stored script.

10. The method of claim 8 wherein further comprising the steps of:

providing graphical user interface language extensions commands to a scripting language; and

passing said generated user input action through said graphical user interface language extensions from said scripting language processor to a language extensions processor.

11. The method of claim 8 further comprising the steps of:

generating a user input action within said computer system responsive to said second element;

monitoring said computer system graphical user interface for an expected third element within a predetermined time interval; and

signaling a failure if said predetermined time interval elapses without detecting said expected third element.

12. The method of claim 8 further comprising the steps of:

depicting said computer system graphical user interface upon a local display including said first element; and

receiving a local user input action within said local display;

wherein said generated user input action emulates said local user input action.

13. The method of claim 8 wherein further comprises comprising the steps of:

providing graphical user interface language extensions commands to a scripting language; and

depicting said computer system graphical user interface upon a local display including said first element;

receiving a local user input action within said local display;

transferring said user input action to a stored script;

passing said generated user input action through said graphical user interface language extensions from said scripting language processor to a language extensions processor wherein said generated user input action emulates said local user input action; and

re-executing said steps of receiving, generating, monitoring and signaling subsequent to said storing step under control of said stored script.

14. A programmerator enabling a local system to remotely operate a computer system through local scripts and selectively respond to changes in graphical displays upon a graphical user interface of said remote computer system, comprising:

a command capture interface that displays a depiction of said remote system graphical user interface display and captures user input made therein;

a command language set that when processed by said local system implements both of user input emulations representative of said captured user input at said remote computer system and image processing of said remote computer

system graphical displays;

a scripting language having scripting commands that control a flow of execution of said local system in combination with said command language set; and

an interface for communicating between said local system and said remote computer system graphical user interface responsive to said command and scripting languages.

15. The programmerator of claim 14 further comprising:

a means for storing said scripting commands;

a means for inserting a command from said command language set into said storing means; and

a means for executing said inserted stored command.